

Alumno: \_\_\_\_\_

Curso: \_\_\_\_\_

## SISTEMAS DE CONTROL

### Microcontroladores con Crocodile technology 3D

#### Contenido

SISTEMAS DE CONTROL .....	1
MICROCONTROLADORES .....	1
CROCODILE TECHNOLOGY 3D.....	2
Ejemplo 1: Luz controlada por interruptor.....	4
Ejemplo 2: Alarma suena 5 veces .....	5
Ejemplo 3: Máquina dispensadora de refrescos .....	6
Ejemplo 4: Lavadora de pago .....	6
Ejemplo 5: Coche sencillo .....	7
Ejemplo 6: Cuadrados .....	8
Desafío 7: Control de una oficina .....	9
Desafío 8: Control de una oficina 2 .....	10
Desafío 9: Código .....	10
Desafío 10: Ladrones .....	10

#### SISTEMAS DE CONTROL

Nuestro **cerebro** es un sistema de control y procesamiento. Recibe **señales de entrada** de todos los sentidos, luego **procesa la información** y toma decisiones **enviando órdenes** a los músculos para realizar un trabajo. Podemos distinguir en este trabajo de nuestro cuerpo tres bloques:

- Entrada de información
- Procesamiento de la información
- Salida de órdenes

En los sistemas electrónicos también trabajamos con los mismos tres bloques con los llamados **microcontroladores** (como es la placa de control de Arduino).

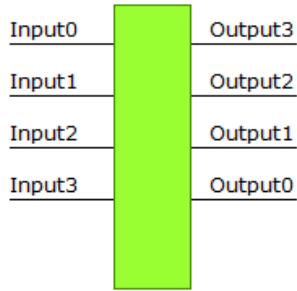
#### MICROCONTROLADORES

Son componentes electrónicos que integran en un solo chip los siguientes elementos:

- Procesador
- Memoria de programa, FLASH
- Memoria de datos (RAM)
- Puertos de Entrada/Salida
- Reloj
- Módulos para controlar periféricos



Un **microcontrolador** típico es el que se muestra a continuación:



INPUT = Entradas de información, se conectan a SENSORES.

OUTPUT = Salidas de órdenes, se conectan a MOTORES, Lámparas, Zumbadores, ...

El microcontrolador mostrado tiene 4 entradas y 4 salidas, esto es variable en otros modelos.

**IDEA principal:** el comportamiento de un microcontrolador, es decir la respuesta de los motores, luces, etc. a los cambios detectados por los sensores, lo diseñamos en un programa que debemos cargar sobre el microcontrolador. Cambiando el programa, cambiamos el comportamiento del microcontrolador.

El modo en que vamos a ESCRIBIR los programas es mediante **diagramas de flujo**, que muestran una secuencia de operaciones ejecutadas de forma ordenada para conseguir un fin, esto es un algoritmo, que puede resolver un problema específico.

Vamos a trabajar con el simulador **Crocodile Technology 3D**.

## CROCODILE TECHNOLOGY 3D

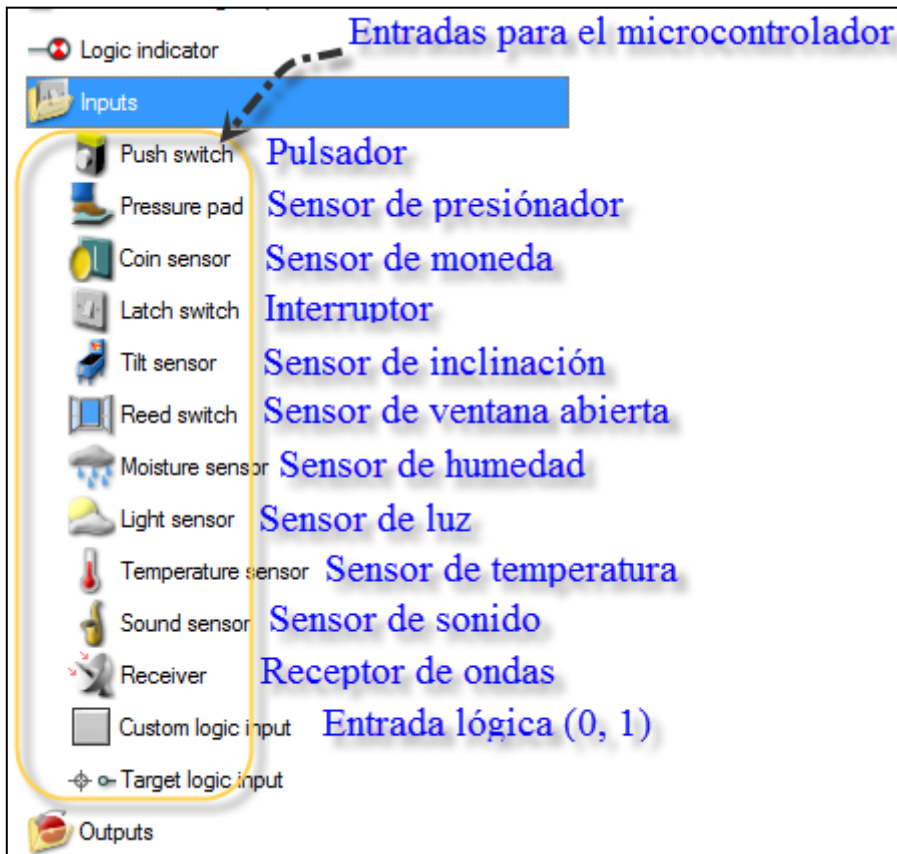
Con este simulador de electrónica podemos realizar las dos partes de nuestros proyectos:

- **Hardware:** montaje de sensores y actuadores en un microcontrolador
- **Software:** mediante diagramas de flujo diseñar el funcionamiento del microcontrolador.

Empezaremos con el **hardware**: todos los componentes que podemos conectar a un microcontrolador son de 2 tipos, elementos de entrada de información o sensores (**INPUT**) y elementos de salida o actuadores (**OUTPUT**).

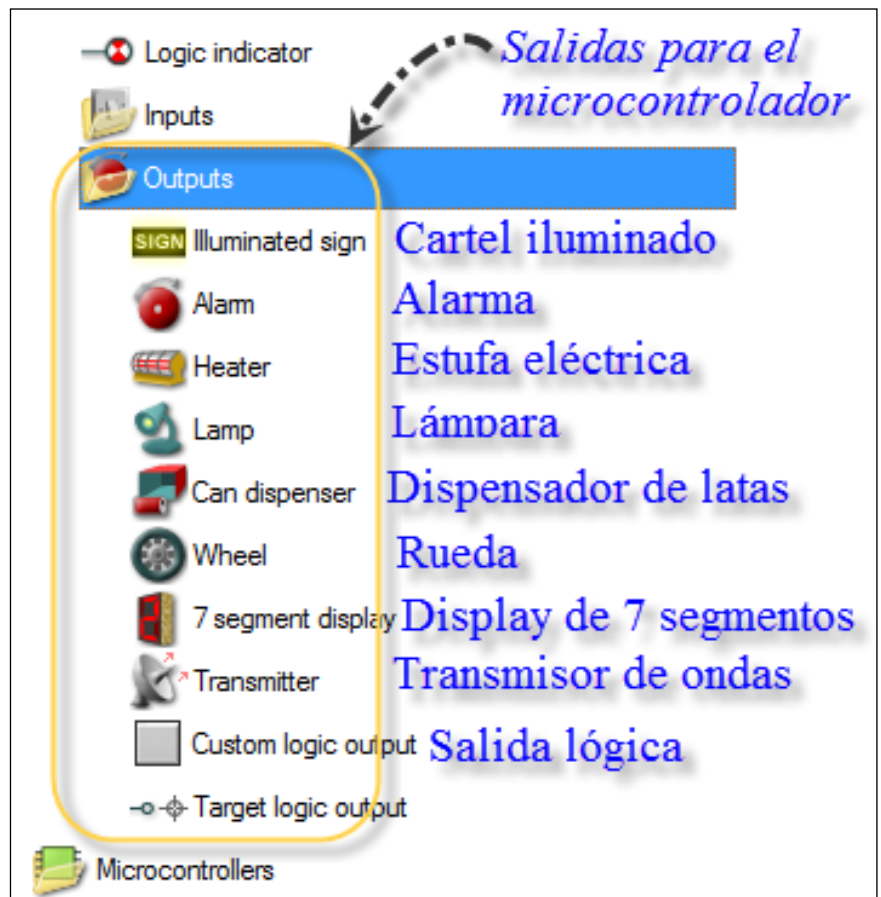
Para encontrar ambas familias abrimos **Crocodile Tech 3D** y entramos en la biblioteca de componentes electrónicos (**Parts Library / Electronics**), vemos organizados por familias, y entre ellas, las entradas y salidas digitales (**Digital Inputs & Outputs**).



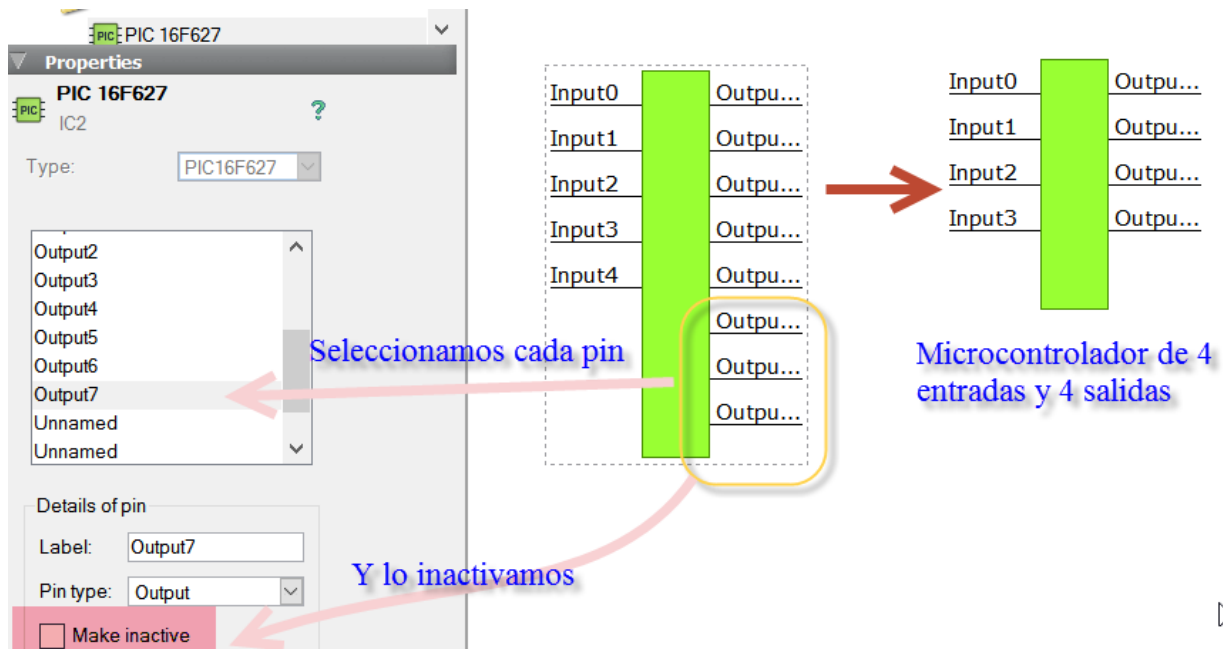


Podemos trabajar con **elementos de entrada**, como son los sensores, abriendo su caja **INPUTS**

También tenemos una gran oferta en elementos **actuadores** o de Salida (OUTPUT)



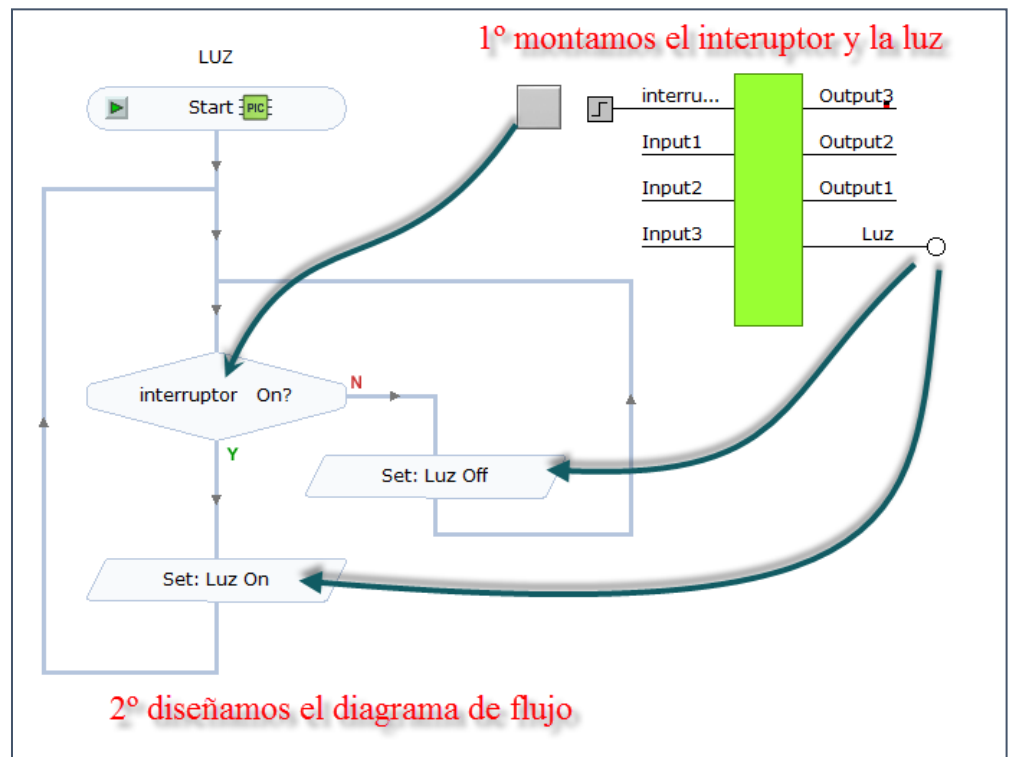
Por último, en cuanto a los **microcontroladores**, usaremos cualquiera de la lista disponible y, si tiene demasiadas patillas, podemos eliminarlas inactivándolas desde el menú propiedades.



Ahora que tenemos ya todos los elementos de hardware vamos a crear nuestro primer algoritmo:

### Ejemplo 1: Luz controlada por interruptor

- **Funcionamiento:** cuando se acciona el interruptor se enciende la luz.
- **Entrada:** interruptor (entrada lógica)
- **Salida:** luz (indicador lógico)



### Explicación del diagrama de flujo: LUZ

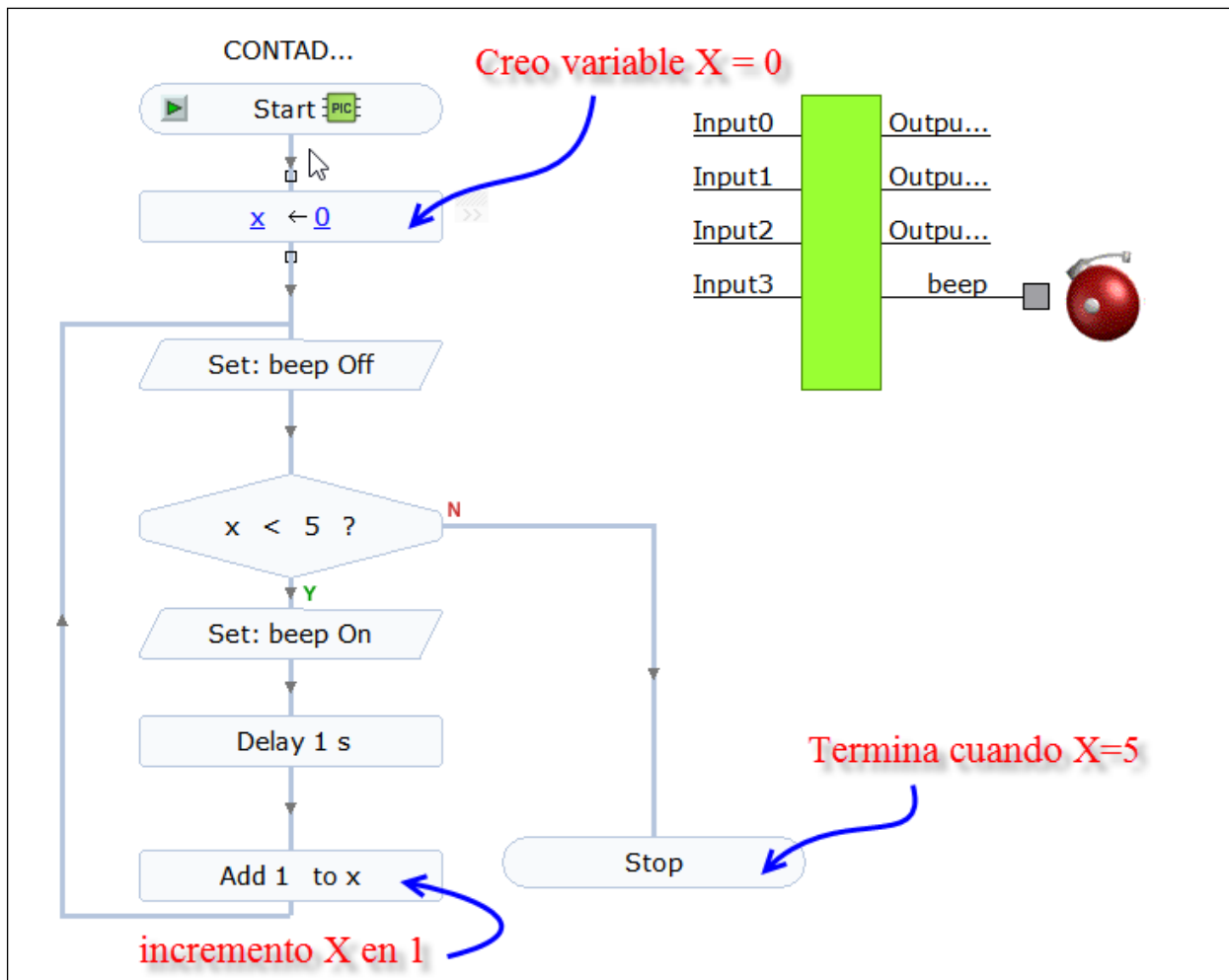
1. Start = empieza el programa
2. interruptor On? = Comprueba si el interruptor ha sido accionado.
3. SET Luz On = Enciende la luz, en caso que se haya accionado el interruptor.
4. SET Luz Off = Apaga la luz, en caso que **NO** se haya accionado el interruptor.
5. Tanto se haya encendido como apagado la luz, las flechas del flujo del programa ascienden hasta justo debajo de START para continuar con el programa en un lazo infinito...

**Desafío uno:** cambia el interruptor por un pulsador y temporiza la luz encendida a 5 segundos cada vez que se presiona el pulsador.

*Pista: para controlar el tiempo en segundos usa la caja DELAY.*

### Ejemplo 2: Alarma suena 5 veces

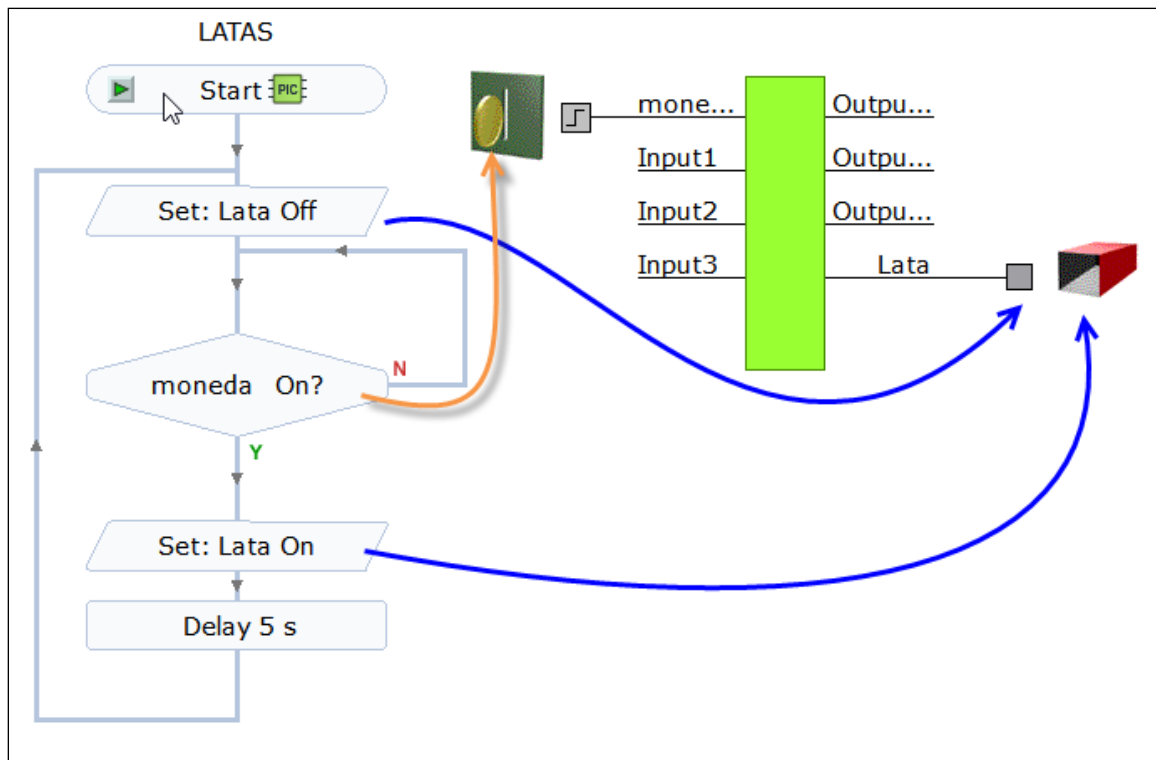
- **Funcionamiento:** una alarma da 5 pitidos de 1 segundo de duración cada uno y luego para.
- Entrada: nada
- **Salida:** alarma
- Necesitaremos crear una variable **x**, luego preguntar si el valor de la variable es menor de 5, si es así sonará *Beep* durante 1 s. A continuación, hay que incrementar la variable **x** en 1 y repetirlo 5 veces.



**Desafío dos:** añade un sensor de humedad de forma que sólo cuando llueva la alarma avise 5 veces.

**Ejemplo 3: Máquina dispensadora de refrescos**

- **Funcionamiento:** cuando se introduce la moneda se dispensará un refresco y esperará 5 segundos hasta estar preparada de nuevo.
- **Entrada:** Sensor de moneda
- **Salida:** dispensador de latas



**Desafío tres:** añade una luz que se encienda durante los 5 segundos que la máquina tarda en estar preparada para dispensar de nuevo refrescos.

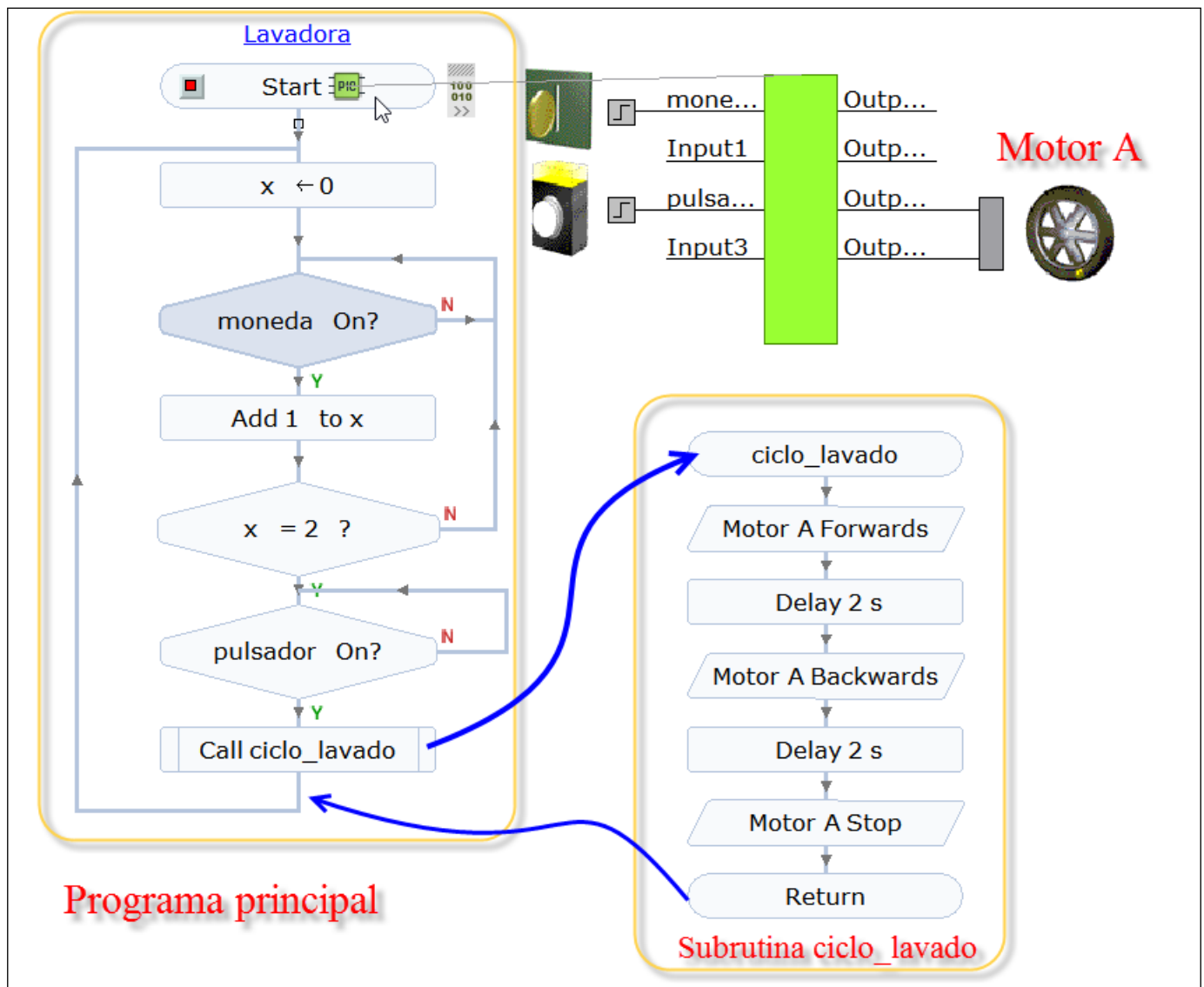
**Ejemplo 4: Lavadora de pago**

- **Funcionamiento:** cuando se introducen 2 monedas y se presiona el pulsador de inicio la lavadora hace este ciclo de lavado: gira en un sentido 2 segundos, luego invierte el giro durante otros 2 segundos y termina.
- **Entrada:** Sensor de moneda y pulsador
- **Salida:** rueda (imaginamos que es el tambor de la lavadora)
- En este programa usamos **subrutinas**, que son programas dentro de programas.

**¿Qué debes saber de las subrutinas?**

1. Permiten separar partes pequeñas del programa principal. Así es más fácil entenderlo y detectar errores.
2. Siempre se llaman desde el programa principal mediante una orden CALL (llamar) con el nombre de la subrutina.
3. Siempre terminan con la orden RETURN para regresar al programa principal.



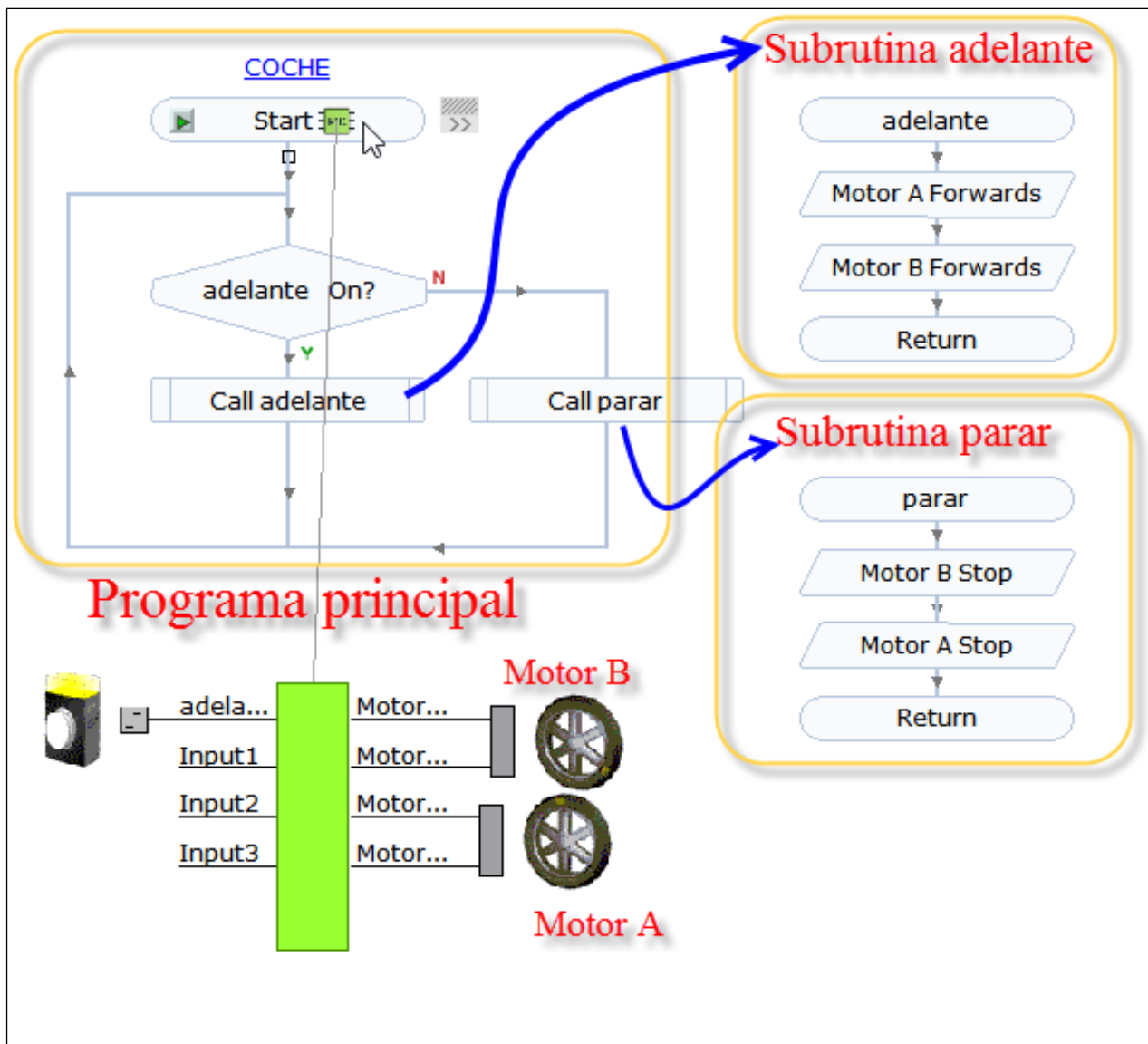


**Desafío cuatro:** añade dos luces que se enciendan cada una con un sentido de giro del tambor de la lavadora.

### Ejemplo 5: Coche sencillo

- **Funcionamiento:** nuestro coche tiene 2 ruedas traseras que puedo controlar y una rueda loca delantera que no se controla. Cuando presiono el pulsador de marcha adelante las dos ruedas giran en un mismo sentido (supondré que es hacia adelante). Cuando NO presiono el mando ambas ruedas se paran.
- **Entrada:** 1 Pulsador (para marcha adelante)
- **Salida:** 2 ruedas (son las dos ruedas traseras, derecha e izquierda, llamadas motor A y motor B)
- En este programa usamos de nuevo **subrutinas**.



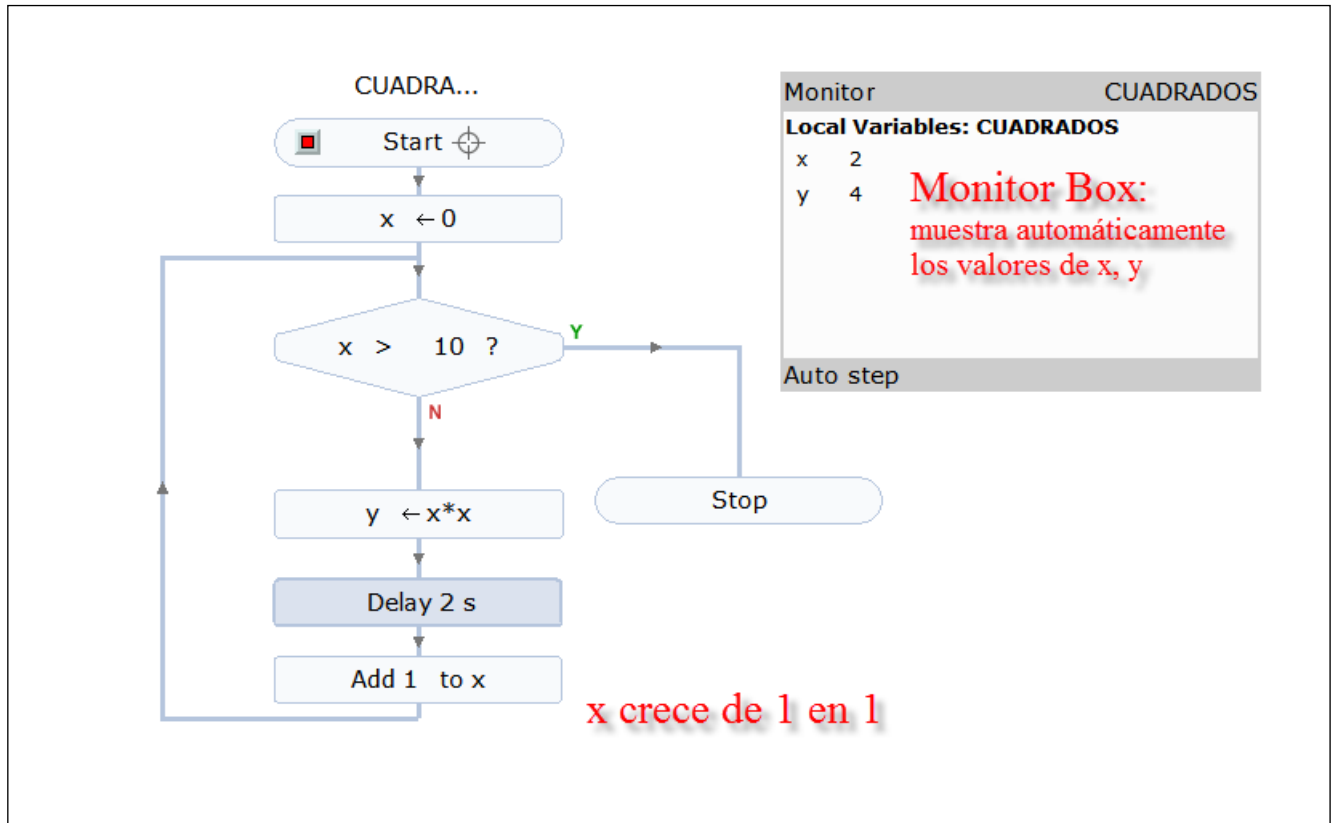


**Desafío cinco:** Añade un segundo mando para ir marcha atrás.

### Ejemplo 6: Cuadrados

- **Funcionamiento:** Diseñar un programa que calcule los cuadrados de los diez primeros números enteros.
- Mostramos los valores en la caja “**Monitor Box**” que se encuentra en FlowChart / Programming Environment / Monitor Box.
- No hace falta **microcontrolador**

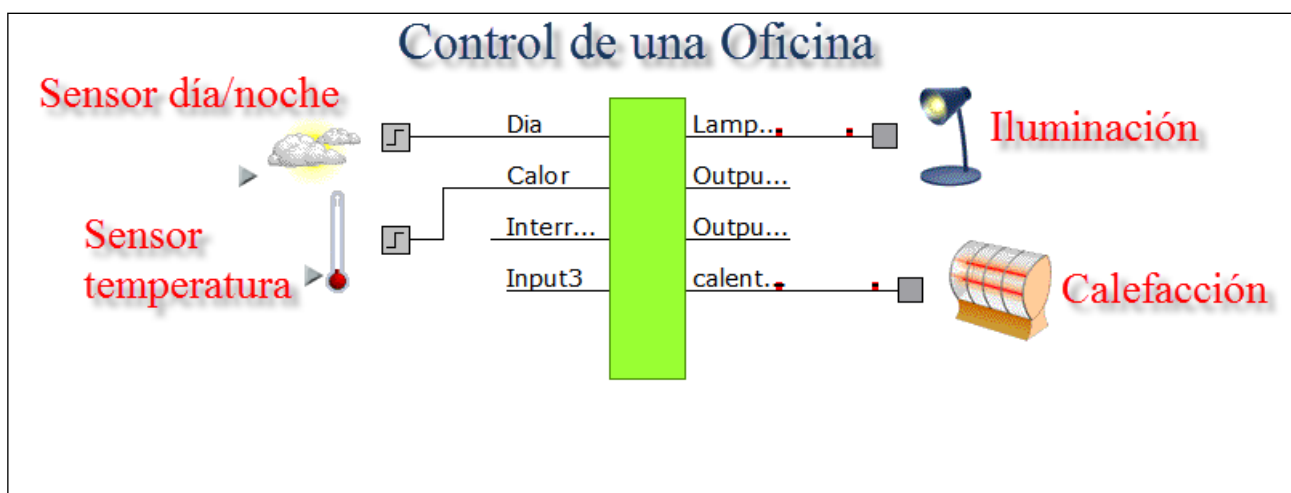




**Desafío seis:** Añadir al diagrama anterior un microcontrolador con un botón para poder pausar el programa cuando queramos. Al volver a pulsar el botón se reanudan los cálculos donde los dejamos.

### Desafío 7: Control de una oficina

- **Funcionamiento:** Las luces se encenderán automáticamente por las noches. La calefacción se pondrá en funcionamiento cuando haga frío y además sea de día.
- **Entrada:** Sensores de luz y de temperatura
- **Salida:** lámpara (simula la iluminación de la oficina y un calentador).
- En este programa debemos usar de nuevo **subrutinas**.

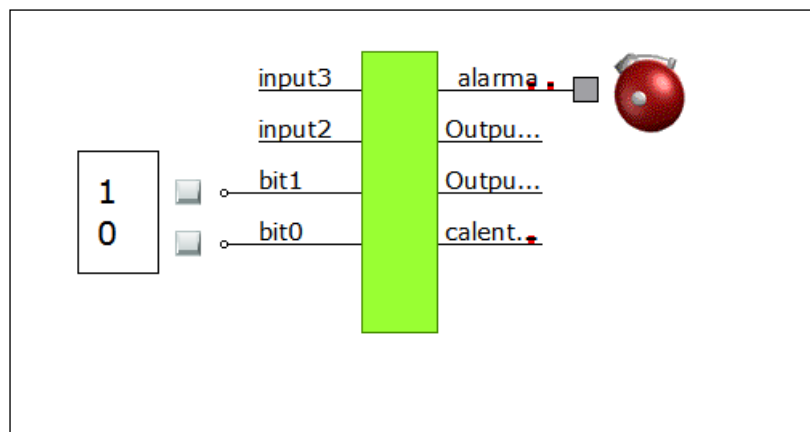


### Desafío 8: Control de una oficina 2

- **Funcionamiento:** Igual que el desafío 6, pero añadimos un interruptor para que se enciendan la iluminación manualmente cuando queramos.
- **Entrada:** Sensores de luz y de temperatura, e interruptor.
- **Salida:** lámpara (simula la iluminación de la oficina y un calentador).
- En este programa debemos usar de nuevo **subrutinas**.

### Desafío 9: Código

- **Funcionamiento:** la alarma empieza sonando fuerte y sólo PARA de SONAR cuando se introduce el siguiente código de bits: bit0 = 0, bit1 = 1.
- **Entrada:** 2 Entradas lógicas.
- **Salida:** Alarma.



### Desafío 10: Ladrones

- **Funcionamiento:** Construir una alarma activada por una alfombrilla de presión. Cuando el sensor de presión detecta a un intruso una luz (Logic indicator) parpadea 5 veces, luego suena una alarma que para sólo cuando se introduce el código correcto (2 Bits: bit0 = 1, bit1 = 1).
- **Entrada:** Sensor de presión, 2 entradas lógicas.
- **Salida:** Alarma, indicador lógico.

